

CLAIMS

What is claimed is:

1. A method in a data processing system having a plurality of elements, each element having corresponding code, the method comprising the steps of:

5 receiving a request to form a link;
receiving an indication of a first of the plurality of elements;
receiving an indication of a second of the plurality of elements; and
in response to receiving the request, the indication of the first element, and the indication of the second element, adding new code to the first element to reflect the link to the second element.

10 2. The method of claim 1, further comprising the step of displaying a graphical representation of the code associated with the first element.

15 3. The method of claim 2, further comprising the step of modifying the graphical representation of the code associated with the first element to reflect the link to the second element.

4. The method of claim 1, wherein the step of adding new code to the first element comprises the steps of:

20 determining whether linking the first element to the second element would violate a predefined rule; and
when it is determined that linking the first element to the second element would not violate a predefined rule,
adding the new code to the first element to form the link to the second element.

5. The method of claim 4, wherein the step of determining whether linking the first element to the second element would violate a predefined rule comprises the steps of:

determining whether the first element is a class and whether the second element is another class; and

5 when it is determined that the first element is the class and that the second element is the other class,

identifying the link from the first element to the second element as an inheritance link.

6. The method of claim 5, further comprising the step of identifying a link error

10 when it is determined that the first element is the class and that the second element is not the other class.

7. The method of claim 4, wherein the step of determining whether linking the first element to the second element would violate a predefined rule comprises the steps of:

determining whether the first element is a class and whether the second element is an interface; and

15 when it is determined that the first element is the class and that the second element is the interface,

identifying the link from the first element to the second element as an implementation link.

20 8. The method of claim 7, further comprising the step of identifying a link error when it is determined that the first element is the class and that the second element is not the interface.

9. The method of claim 4, wherein the step of determining whether linking the first element to the second element would violate a predefined rule comprises the steps of:

- determining whether the first element is an interface and the second element is another interface; and
- 5 when it is determined that the first element is the interface and the second element is the other interface,
 - identifying the link from the first element to the second element as an inheritance link.

10. The method of claim 9, further comprising the step of identifying a link error

10 when it is determined that the first element is the interface and the second element is not the other interface.

11. A method in a data processing system having a plurality of elements, each element having corresponding code, the method comprising the steps of:

- receiving a request to form a link;
- receiving an indication of a first of the plurality of elements;
- 5 receiving an indication of a second of the plurality of elements;
- determining whether linking the first element to the second element would violate a predefined rule; and
- when it is determined that linking the first element to the second element would not violate a predefined rule,

10 adding new code to the first element to reflect the link to the second element.

12. The method of claim 11, further comprising the step of displaying a graphical representation of the code associated with the first element.

13. The method of claim 12, further comprising the step of modifying the graphical representation of the code associated with the first element to reflect the link to the 15 second element.

14. The method of claim 11, wherein the step of determining whether linking the first element to the second element would violate a predefined rule comprises the steps of:

- determining whether the first element is a class and whether the second element is another class; and

20 when it is determined that the first element is the class and that the second element is the other class,

identifying the link from the first element to the second element as an inheritance link.

15. The method of claim 11, wherein the step of determining whether linking the first element to the second element would violate a predefined rule comprises the steps of:

determining whether the first element is a class and whether the second element is an interface; and

5 when it is determined that the first element is the class and that the second element is the interface,

identifying the link from the first element to the second element as an implementation link.

16. The method of claim 11, wherein the step of determining whether linking the 10 first element to second element would violate a predefined rule comprises the steps of:

determining whether the first element is an interface and the second element is another interface; and

when it is determined that the first element is the interface and the second element is the other interface,

15 identifying the link from the first element to the second element as an inheritance link.

17. A method in a data processing system having a plurality of elements, each element having corresponding code, the method comprising the steps of:

displaying a graphical representation of the code associated with a first of the plurality of elements and a graphical representation of the code associated with a second of the plurality of elements;

5 receiving a request to form a link from the first element to the second element;

determining whether the first element is a class and whether the second element is another class; and

10 when it is determined that the first element is the class and that the second element is the other class,

identifying the link from the first element to the second element as an inheritance link;

adding new code to the first element to reflect the link to the second element; and

15 modifying the graphical representation of the code associated with the first element to reflect the link to the second element.

18. The method of claim 17, further comprising the steps of:

when it is determined that the first element is the class and that the second element is not the other class,

20 determining whether the second element is an interface; and

when it is determined that the second element is the interface,

identifying the link from the first element to the second element as an implementation link.

19. The method of claim 17, further comprising the steps of:
when it is determined that the first element is not the class and that the second
element is not the other class,
determining whether the first and the second elements are interfaces; and
when it is determined that the first and second elements are interfaces,
identifying the link from the first element to the second element as an
inheritance link.

20. A method in a data processing system having a plurality of elements and having a link between two of the plurality of elements, wherein each element has corresponding code and the linked elements include a source and a destination, the method comprising the steps of:

- 5 receiving a selection of one of the linked elements;
- receiving an identification of another of the plurality of elements that is different than the linked elements;
- determining whether the selected element is the destination; and
- when it is determined that the selected element is the destination,
 - 10 modifying the corresponding code of the other element to reflect a new link between the other element and the destination element.

21. The method of claim 20, wherein the modifying step further includes the step of modifying the corresponding code of the source to reflect the removal of the link between the source and the destination.

22. The method of claim 21, further comprising the step of displaying a graphical representation of the corresponding code of the source.

23. The method of claim 22, further comprising the step of modifying the graphical representation of the corresponding code of the source to reflect the removal of the link between the source and the destination.

20 24. The method of claim 20, further comprising the step of displaying a graphical representation of the corresponding code of the other element.

25. The method of claim 24, further comprising the step of modifying the graphical representation of the corresponding code of the other element to reflect the new link between the other element and the destination element.

26. The method of claim 20, wherein the modifying step includes the steps of:
determining whether linking the other element to the destination would violate a predefined rule; and
when it is determined that linking the other element to the destination would not violate a predefined rule,
5 modifying the corresponding code of the source to reflect the removal of the link between the source and the destination; and
adding new code to the corresponding code of the other element to reflect the new link between the other element and the destination element.

10 27. The method of claim 26, wherein the step of determining whether linking the other element to the destination would violate a predefined rule, comprises the steps of:
determining whether the other element is a class and whether the destination is another class; and
when it is determined that the other element is the class and that the destination is the other class,
15 identifying the new link between the other element and the destination as an inheritance link.

20 28. The method of claim 26, wherein the step of determining whether linking the other element to the destination would violate a predefined rule, comprises the steps of:
determining whether the other element is a class and whether the destination is an interface; and
when it is determined that the other element is the class and that the destination is the interface,
25 identifying the new link between the other element and the destination as an implementation link.

29. The method of claim 28, further comprising the step of identifying a link error when it is determined that the other element is the class and that the destination is not the interface.

30. The method of claim 26, wherein the step of determining whether linking the other element to the destination would violate a predefined rule, comprises the steps of:

determining whether the other element is an interface and the destination is another interface; and

5 when it is determined that the other element is the interface and the destination is the other interface,

identifying the new link between the other element and the destination as an inheritance link.

31. The method of claim 30, further comprising the step of identifying a link 10 error when it is determined that the other element is not the interface.

32. The method of claim 30, further comprising the step of identifying a link error when it is determined that the destination is not the other interface.

33. The method of claim 20 further comprising the step of:
when it is determined that the selected element is the source,

15 modifying the corresponding code of the source to reflect a new link between the source and the other element.

34. The method of claim 33, further comprising the steps of:
when it is determined that the selected element is the source,

20 determining whether linking the source to the other element would violate a predefined rule; and

when it is determined that linking the source to the other element would not violate a predefined rule,

modifying the corresponding code of the source to reflect the removal of the link between the source and the destination; and

25 adding new code to the corresponding code of the source to reflect the new link to the other element.

35. The method of claim 34, further comprising the step of displaying a graphical representation of the code of the source to reflect the removal of the link and to reflect the new link to the other element when it is determined that the selected element is the source.

36. The method of claim 34, wherein the step of determining whether linking the source to the other element would violate a predefined rule, comprises the steps of:

determining whether the source is a class and whether the other element is another class; and

when it is determined that the source is the class and that the other element is the other class,

10 identifying the new link between the source and the other element as an inheritance link.

37. The method of claim 34, wherein the step of determining whether linking the source to the other element would violate a predefined rule, comprises the steps of:

determining whether the source is a class and whether the other element is an interface; and

when it is determined that the source is the class and that the other element is the interface,

identifying the new link from the source to the other element as an implementation link.

20 38. The method of claim 37, further comprising the step of identifying a link error when it is determined that the other element is not the interface.

39. The method of claim 34, wherein the step of determining whether linking the source to the other element would violate a predefined rule, comprises the steps of:

determining whether the source is an interface and the other element is another interface; and

5 when it is determined that the source is the interface and the other element is the other interface,

identifying the new link between the source and the other element as an inheritance link.

40. The method of claim 39, further comprising the step of identifying a link 10 error when it is determined that the source is not the interface.

41. The method of claim 39, further comprising the step of identifying a link error when it is determined that the other element is not the other interface.

42. A method in a data processing system having a plurality of elements and having a link between two of the plurality of elements, wherein each element has corresponding code and the linked elements include a source and a destination, the method comprising the steps of:

- 5 receiving an identification of the link;
- receiving a selection of one of the linked elements;
- receiving an identification of another of the plurality of elements that is different than the linked elements;
- determining whether the selected element is the source; and
- 10 when it is determined that the selected element is the source,
 - modifying the corresponding code of the source to reflect a new link between the source and the other element.

43. The method of claim 42, further comprising the step of displaying a graphical representation of the corresponding code of the source and a graphical representation of the 15 corresponding code of the other element.

44. The method of claim 43, further comprising the step of modifying the graphical representation of the corresponding code of the source to reflect the removal of the link when it is determined that the selected element is the source.

45. The method of claim 42, further comprising the steps of:
when it is determined that the selected element is the source,
determining whether linking the source to the other element would violate a
predefined rule; and
5 when it is determined that linking the source to the other element would not
violate a predefined rule,
modifying the corresponding code of the source to reflect the removal
of the link between the source and the destination; and
adding new code to the corresponding code of the source to reflect the
new link to the other element.

10
46. The method of claim 45, further comprising the step of displaying a graphical
representation of the code of the source to reflect the removal of the link and to reflect the
new link to the other element when it is determined that the selected element is the source.

15
47. The method of claim 45, wherein the step of determining whether linking the
source to the other element would violate a predefined rule, comprises the steps of:
determining whether the source is a class and whether the other element is another
class; and
when it is determined that the source is the class and that the other element is the
other class,
20 identifying the new link between the source and the other element as an
inheritance link.

48. The method of claim 45, wherein the step of determining whether linking the source to the other element would violate a predefined rule, comprises the steps of:

determining whether the source is a class and whether the other element is an interface; and

5 when it is determined that the source is the class and that the other element is the interface,

identifying the new link from the source to the other element as an implementation link.

49. The method of claim 48, further comprising the step of identifying a link 10 error when it is determined that the other element is not the interface.

50. The method of claim 45, wherein the step of determining whether linking the source to the other element would violate a predefined rule, comprises the steps of:

determining whether the source is an interface and the other element is another interface; and

when it is determined that the source is the interface and the other element is the other interface,

identifying the new link between the source and the other element as an inheritance link.

51. The method of claim 50, further comprising the step of identifying a link 20 error when it is determined that the source is not the interface.

52. The method of claim 50, further comprising the step of identifying a link error when it is determined that the other element is not the other interface.

53. The method of claim 42 further comprising the step of:

when it is determined that the selected element is the destination,

25 modifying the corresponding code of the other element to reflect a new link between the other element and the destination element.

54. The method of claim 53, wherein the modifying step further includes the steps of:

determining whether linking the other element to the destination would violate a predefined rule; and

5 when it is determined that linking the other element to the destination would not violate a predefined rule,

modifying the corresponding code of the source to reflect the removal of the link between the source and the destination; and

10 adding new code to the corresponding code of the other element to reflect a new link between the other element and the destination element.

55. The method of claim 54, wherein the step of determining whether linking the other element to the destination would violate a predefined rule, comprises the steps of:

determining whether the other element is a class and whether the destination is another class; and

15 when it is determined that the other element is the class and that the destination is the other class,

identifying the new link between the other element and the destination as an inheritance link.

56. The method of claim 54, wherein the step of determining whether linking the

20 other element to the destination would violate a predefined rule, comprises the steps of:

determining whether the other element is a class and whether the destination is an interface; and

when it is determined that the other element is the class and that the destination is the interface,

25 identifying the new link between the other element and the destination as an implementation link.

57. The method of claim 56, further comprising the step of identifying a link error when it is determined that the destination is not the interface.

58. The method of claim 54, wherein the step of determining whether linking the other element to the destination would violate a predefined rule, comprises the steps of:

determining whether the other element is an interface and the destination is another interface; and

5 when it is determined that the other element is the interface and the destination is the other interface,

identifying the new link between the other element and the destination as an inheritance link.

59. The method of claim 58, further comprising the step of identifying a link 10 error when it is determined that the other element is not the interface.

60. The method of claim 58, further comprising the step of identifying a link error when it is determined that the destination is not the other interface.

61. A method in a data processing system having a plurality of elements and having a link between two of the plurality of elements, wherein each element has corresponding code and the linked elements include a source and a destination, the method comprising the steps of:

- 5 displaying a graphical representation of the corresponding code of each of the plurality of elements;
- receiving a selection of one of the linked elements;
- receiving an identification of another of the plurality of elements that is different than the linked element;
- 10 determining whether the selected element is the destination; and
- when it is determined that the selected element is the destination,
 - determining whether the other element is a class and whether the destination is another class; and
 - when it is determined that the other element is the class and that the destination is the other class,
 - identifying a new link from the other element to the destination as an inheritance link;
 - removing a portion of the corresponding code of the source that reflects the link between the source and the destination;
 - adding new code to corresponding code of the other element to reflect the new link between the other element and the destination;
 - modifying the graphical representation of the corresponding code of the source to reflect the removal of the link; and
 - modifying the graphical representation of the corresponding code of the other element to reflect the new link.
- 20
- 25

62. The method of claim 61, further comprising the steps of:
when it is determined that the other element is the class and that the destination is not
the other class,

5 determining whether the destination is an interface; and

when it is determined that the destination is the interface,

10 identifying the new link between the other element and the destination
as an implementation link.

63. The method of claim 61, further comprising the steps of:
when it is determined that the other element is not the class and that the destination is
not the other class,

15 determining whether the other element is an interface and whether the
destination is another interface; and

when it is determined that the other element is the interface and that the
destination is the other interface,

20 identifying the new link between the other element and the destination
as an inheritance link.

64. The method of claim 61, further comprising the steps of:
when it is determined that the selected element is the source,

25 determining whether the source is a class and whether the other element is
another class; and

when it is determined that the source is the class and that the other element is
the other class,

identifying the new link between the source and the other element as
an inheritance link;

removing a portion of the corresponding code of the source that
reflects the link between the source and the destination; and

25 adding new code to corresponding code of the source to reflect the
new link between the source and the other element.

65. The method of claim 64, further comprising the step of modifying a graphical representation of the corresponding code of the source to reflect the removal of the link and to reflect the new link when it is determined that linking the source to the other element would not violate a predefined rule.

5 66. The method of claim 64, further comprising the steps of:
when it is determined that the source is the class and that the other element is not the other class,

determining whether the other element is in an interface; and
when it is determined that the other element is the interface,

10 identifying the new link between the source and the other element as
an implementation link.

67. The method of claim 64, further comprising the steps of:
when it is determined that the source is not the class and that the other element is not the other class,

determining whether the source is an interface and the other element is
another interface; and
when it is determined that the source is the interface and the other element is
the other interface,

15 identifying the new link between the source and the other element as
an inheritance link.

20

68. A method in a data processing system having a plurality of elements, the method comprising the steps of:

- receiving an identification of a first of the plurality of elements;
- receiving an identification of a second of the plurality of elements;
- 5 receiving an indication that the first element is to be included in the second element;
- determining whether the first element is a class and whether the second element is another class; and
- when it is determined that the first element is the class and that the second element is the other class,

10 transferring code corresponding to the first element into the second element.

69. The method of claim 68, wherein the method further comprises the step of displaying a graphical representation of the code of the first element and a graphical representation of the code of the second element.

70. The method of claim 69, further comprising the step of modifying the graphical representation of the code of the second element to reflect the transfer of the code corresponding to the first element into the second element.

71. The method of claim 68, wherein the step of transferring code comprises the steps of:

- 20 removing the code corresponding to the first element from a file;
- placing the code corresponding to the first element within the code corresponding to the second element; and
- deleting the file.

72. The method of claim 68, wherein the method further comprises the steps of:
when it is determined that the first element is the class and that the second element is
not the other class,

5 determining whether the second element is a package; and

when it is determined that the second element is a package,

moving a file that includes code corresponding to the first element to
a directory associated with the second element.

73. A method in a data processing system having a plurality of elements, each element having corresponding code, the method comprising the steps of:

displaying a graphical representation of the code of a first of the plurality of elements and a graphical representation of the code of a second of the plurality of elements;

5 receiving an indication that the first element is to be included in the second element; determining whether the first element is a class and whether the second element is another class; and

when it is determined that the first element is the class and that the second element is the other class,

10 transferring code corresponding to the first element into the second element; and

modifying a graphical representation of the code of the second element to reflect the transfer of the first element into the second element.

74. The method of claim 73, wherein the step of transferring code comprises the

15 steps of:

removing the code corresponding to the first element from a file; placing the code corresponding to the first element within code corresponding to the second element; and deleting the file.

75. A method in a data processing system having a plurality of elements, each element having corresponding code, wherein code corresponding to a first of the plurality of elements is nested in the code corresponding to a second of the plurality of elements, the method comprising the steps of:

5 receiving an indication that the first element is to be removed from the second element;

determining whether the first element is a class and whether the second element is another class; and

when it is determined that the first element is the class and that the second element is the other class,

10 removing code corresponding to the first element from the second element.

76. The method of claim 75, further comprising the step of placing the code corresponding to the first element into a file.

77. The method of claim 76, further comprising the step of displaying a graphical representation of the code corresponding to the first element.

78. The method of claim 75, further comprising the step of modifying a graphical representation of the code corresponding to the second element to reflect the removal of the first element from the second element.

79. The method of claim 75, further comprising the steps of:

20 when it is determined that the first element is the class and that the second element is not the other class,

determining whether the second element is a package; and

when it is determined that the second element is the package,

25 removing a first file that includes code corresponding to the first element from a directory associated with the second element to another directory.

80. A method in a data processing system having a plurality of elements, wherein a first of the plurality of elements is nested within a second of the plurality of elements, the method comprising the steps of:

5 receiving an indication that the first element is to be removed from the second element;

determining whether the first element is a class and whether the second element is another class; and

when it is determined that the first element is the class and that the second element is the other class,

10 removing code corresponding to the first element from the second element;

placing the code corresponding to the first element into a file; and

displaying a graphical representation of the code corresponding to the first element.

81. The method of claim 80, further comprising the step of modifying a graphical representation of the code corresponding to the second element to reflect the removal of the first element from the second element.

82. The method of claim 80, further comprising the steps of:

when it is determined that the first element is the class and the second element is not the other class,

20 determining whether the second element is a package; and

when it is determined that the second element is the package,

removing a first file that includes code corresponding to the first element from a directory associated with the second element.

83. A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of elements, each element having corresponding code, the method comprising the steps of:

- receiving a request to form a link;
- 5 receiving an indication of a first of the plurality of elements;
- receiving an indication of a second of the plurality of elements; and
- in response to receiving the request, the indication of the first element, and the indication of the second element, adding new code to the first element to reflect the link to the second element.

10 84. The computer-readable medium of claim 83, wherein the method further comprises the step of displaying a graphical representation of the code associated with the first element.

15 85. The computer-readable medium of claim 84, wherein the method further comprises the step of modifying the graphical representation of the code associated with the first element to reflect the link to the second element.

86. The computer-readable medium of claim 83, wherein the step of adding new code to the first element comprises the steps of:

- 20 determining whether linking the first element to the second element would violate a predefined rule; and
- when it is determined that linking the first element to the second element would not violate a predefined rule,
- adding the new code to the first element to form the link to the second element.

87. The computer-readable medium of claim 86, wherein the step of determining whether linking the first element to the second element would violate a predefined rule comprises the steps of:

5 determining whether the first element is a class and whether the second element is another class; and

when it is determined that the first element is the class and that the second element is the other class,

identifying the link from the first element to the second element as an inheritance link.

10 88. The computer-readable medium of claim 87, wherein the method further comprises the step of identifying a link error when it is determined that the first element is the class and that the second element is not the other class.

15 89. The computer-readable medium of claim 86, wherein the step of determining whether linking the first element to the second element would violate a predefined rule comprises the steps of:

determining whether the first element is a class and whether the second element is an interface; and

when it is determined that the first element is the class and that the second element is the interface,

20 identifying the link from the first element to the second element as an implementation link.

90. The computer-readable medium of claim 89, wherein the method further comprises the step of identifying a link error when it is determined that the first element is the class and that the second element is not the interface.

91. The computer-readable medium of claim 86, wherein the step of determining whether linking the first element to the second element would violate a predefined rule comprises the steps of:

5 determining whether the first element is an interface and the second element is another interface; and

when it is determined that the first element is the interface and the second element is the other interface,

identifying the link from the first element to the second element as an inheritance link.

10 92. The computer-readable medium of claim 91, wherein the method further comprises the step of identifying a link error when it is determined that the first element is the interface and the second element is not the other interface.

93. A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of elements, each element having corresponding code, the method comprising the steps of:

receiving a request to form a link;

5 receiving an indication of a first of the plurality of elements;

receiving an indication of a second of the plurality of elements;

determining whether linking the first element to the second element would violate a predefined rule; and

when it is determined that linking the first element to the second element would not violate a predefined rule,

adding new code to the first element to reflect the link to the second element.

94. The computer-readable medium of claim 93, wherein the method further comprises the step of displaying a graphical representation of the code associated with the first element.

95. The computer-readable medium of claim 94, wherein the method further comprises the step of modifying the graphical representation of the code associated with the first element to reflect the link to the second element.

96. The computer-readable medium of claim 93, wherein the step of determining whether linking the first element to the second element would violate a predefined rule comprises the steps of:

determining whether the first element is a class and whether the second element is another class; and

when it is determined that the first element is the class and that the second element is the other class

25 identifying the link from the first element to the second element as an inheritance link.

97. The computer-readable medium of claim 93, wherein the step of determining whether linking the first element to the second element would violate a predefined rule comprises the steps of:

determining whether the first element is a class and whether the second element is an interface; and

when it is determined that the first element is the class and that the second element is the interface.

identifying the link from the first element to the second element as an implementation link.

98. The computer-readable medium of claim 93, wherein the step of determining whether linking the first element to second element would violate a predefined rule comprises the steps of:

determining whether the first element is an interface and the second element is another interface; and

when it is determined that the first element is the interface and the second element is the other interface.

identifying the link from the first element to the second element as an inheritance link.

99. A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of elements, each element having corresponding code, the method comprising the steps of:

displaying a graphical representation of the code associated with a first of the plurality of elements and a graphical representation of the code associated with a second of the plurality of elements;

receiving a request to form a link from the first element to the second element;

determining whether the first element is a class and whether the second element is another class; and

when it is determined that the first element is the class and that the second element is the other class.

identifying the link from the first element to the second element as an inheritance link;

adding new code to the first element to reflect the link to the second element;
and

modifying the graphical representation of the code associated with the first element to reflect the link to the second element.

100. The computer-readable medium of claim 99, wherein the method further comprises the steps of:

when it is determined that the first element is the class and that the second element is not the other class,

determining whether the second element is an interface; and

when it is determined that the second element is the interface,

identifying the link from the first element to the second element as an

implementation link.

101. The computer-readable medium of claim 99, wherein the method further comprises the steps of:

when it is determined that the first element is not the class and that the second element is not the other class,

determining whether the first and the second elements are interfaces; and

when it is determined that the first and second elements are interfaces,

identifying the link from the first element to the second element as an inheritance link.

102. A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of elements and having a link between two of the plurality of elements, wherein each element has corresponding code and the linked elements include a source and a destination, the 5 method comprising the steps of:

- receiving a selection of one of the linked elements;
- receiving an identification of another of the plurality of elements that is different than the linked elements;
- determining whether the selected element is the destination; and
- when it is determined that the selected element is the destination,
 - modifying the corresponding code of the other element to reflect a new link between the other element and the destination element.

103. The computer-readable medium of claim 102, wherein the modifying step further includes the step of modifying the corresponding code of the source to reflect the removal of the link between the source and the destination. 15

104. The computer-readable medium of claim 103, wherein the method further comprises the step of displaying a graphical representation of the corresponding code of the source.

105. The computer-readable medium of claim 104, wherein the method further 20 comprises the step of modifying the graphical representation of the corresponding code of the source to reflect the removal of the link between the source and the destination.

106. The computer-readable medium of claim 102, wherein the method further comprises the step of displaying a graphical representation of the corresponding code of the other element.

107. The computer-readable medium of claim 106, wherein the method further comprises the step of modifying the graphical representation of the corresponding code of the other element to reflect the new link between the other element and the destination element.

5 108. The computer-readable medium of claim 102, wherein the modifying step includes the steps of:

determining whether linking the other element to the destination would violate a predefined rule; and

when it is determined that linking the other element to the destination would not violate a predefined rule,

modifying the corresponding code of the source to reflect the removal of the link between the source and the destination; and

adding new code to the corresponding code of the other element to reflect the new link between the other element and the destination element.

15 109. The computer-readable medium of claim 108, wherein the step of determining whether linking the other element to the destination would violate a predefined rule, comprises the steps of:

determining whether the other element is a class and whether the destination is another class; and

20 when it is determined that the other element is the class and that the destination is the other class,

identifying the new link between the other element and the destination as an inheritance link.

110. The computer-readable medium of claim 108, wherein the step of determining whether linking the other element to the destination would violate a predefined rule, comprises the steps of:

5 determining whether the other element is a class and whether the destination is an interface; and

when it is determined that the other element is the class and that the destination is the interface,

identifying the new link between the other element and the destination as an implementation link.

10 111. The computer-readable medium of claim 110, wherein the method further comprises the step of identifying a link error when it is determined that the other element is the class and that the destination is not the interface.

15 112. The computer-readable medium of claim 108, wherein the step of determining whether linking the other element to the destination would violate a predefined rule, comprises the steps of:

determining whether the other element is an interface and the destination is another interface; and

when it is determined that the other element is the interface and the destination is the other interface,

20 identifying the new link between the other element and the destination as an inheritance link.

113. The computer-readable medium of claim 112, wherein the method further comprises the step of identifying a link error when it is determined that the other element is not the interface.

25 114. The computer-readable medium of claim 112, wherein the method further comprises the step of identifying a link error when it is determined that the destination is not the other interface.

115. The computer-readable medium of claim 102 wherein the method further comprises the step of:

when it is determined that the selected element is the source,
modifying the corresponding code of the source to reflect a new link between
the source and the other element.

5

116. The computer-readable medium of claim 115, wherein the method further comprises the steps of:

when it is determined that the selected element is the source,
determining whether linking the source to the other element would violate a
predefined rule; and

10

when it is determined that linking the source to the other element would not
violate a predefined rule,

modifying the corresponding code of the source to reflect the removal
of the link between the source and the destination; and

15

adding new code to the corresponding code of the source to reflect the
new link to the other element.

20

117. The computer-readable medium of claim 116, wherein the method further comprises the step of displaying a graphical representation of the code of the source to reflect the removal of the link and to reflect the new link to the other element when it is determined that the selected element is the source.

118. The computer-readable medium of claim 116, wherein the step of determining whether linking the source to the other element would violate a predefined rule, comprises the steps of:

5 determining whether the source is a class and whether the other element is another class; and

when it is determined that the source is the class and that the other element is the other class,

identifying the new link between the source and the other element as an inheritance link.

10 119. The computer-readable medium of claim 116, wherein the step of determining whether linking the source to the other element would violate a predefined rule, comprises the steps of:

determining whether the source is a class and whether the other element is an interface; and

15 when it is determined that the source is the class and that the other element is the interface,

identifying the new link from the source to the other element as an implementation link.

120. The computer-readable medium of claim 119, wherein the method further 20 comprises the step of identifying a link error when it is determined that the other element is not the interface.

121. The computer-readable medium of claim 116, wherein the step of determining whether linking the source to the other element would violate a predefined rule, comprises the steps of:

5 determining whether the source is an interface and the other element is another interface; and

when it is determined that the source is the interface and the other element is the other interface,

identifying the new link between the source and the other element as an inheritance link.

10 122. The computer-readable medium of claim 121, wherein the method further comprises the step of identifying a link error when it is determined that the source is not the interface.

15 123. The computer-readable medium of claim 121, wherein the method further comprises the step of identifying a link error when it is determined that the other element is not the other interface.

124. A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of elements and having a link between two of the plurality of elements, wherein each element has corresponding code and the linked elements include a source and a destination, the 5 method comprising the steps of:

- receiving an identification of the link;
- receiving a selection of one of the linked elements;
- receiving an identification of another of the plurality of elements that is different than the linked elements;
- 10 determining whether the selected element is the source; and
- when it is determined that the selected element is the source,
 - modifying the corresponding code of the source to reflect a new link between the source and the other element.

125. The computer-readable medium of claim 124, wherein the method further 15 comprises the step of displaying a graphical representation of the corresponding code of the source and a graphical representation of the corresponding code of the other element.

126. The computer-readable medium of claim 125, wherein the method further comprises the step of modifying the graphical representation of the corresponding code of the source to reflect the removal of the link when it is determined that the selected element is 20 the source.

127. The computer-readable medium of claim 124, wherein the method further comprises the steps of:

when it is determined that the selected element is the source,

5 determining whether linking the source to the other element would violate a predefined rule; and

when it is determined that linking the source to the other element would not violate a predefined rule,

modifying the corresponding code of the source to reflect the removal of the link between the source and the destination; and

10 adding new code to the corresponding code of the source to reflect the new link to the other element.

128. The computer-readable medium of claim 127, wherein the method further comprises the step of displaying a graphical representation of the code of the source to reflect the removal of the link and to reflect the new link to the other element when it is determined that the selected element is the source.

129. The computer-readable medium of claim 127, wherein the step of determining whether linking the source to the other element would violate a predefined rule, comprises the steps of:

determining whether the source is a class and whether the other element is another 20 class; and

when it is determined that the source is the class and that the other element is the other class,

identifying the new link between the source and the other element as an inheritance link.

130. The computer-readable medium of claim 127, wherein the step of determining whether linking the source to the other element would violate a predefined rule, comprises the steps of:

5 determining whether the source is a class and whether the other element is an interface; and

when it is determined that the source is the class and that the other element is the interface,

identifying the new link from the source to the other element as an implementation link.

10 131. The computer-readable medium of claim 130, wherein the method further comprises the step of identifying a link error when it is determined that the other element is not the interface.

15 132. The computer-readable medium of claim 127, wherein the step of determining whether linking the source to the other element would violate a predefined rule, comprises the steps of:

 determining whether the source is an interface and the other element is another interface; and

 when it is determined that the source is the interface and the other element is the other interface,

20 identifying the new link between the source and the other element as an inheritance link.

133. The computer-readable medium of claim 132, wherein the method further comprises the step of identifying a link error when it is determined that the source is not the interface.

25 134. The computer-readable medium of claim 132, wherein the method further comprises the step of identifying a link error when it is determined that the other element is not the other interface.

135. The computer-readable medium of claim 124, wherein the method further comprises the step of:

when it is determined that the selected element is the destination,
modifying the corresponding code of the other element to reflect a new link
between the other element and the destination element.

5

136. The computer-readable medium of claim 135, wherein the modifying step further includes the steps of:

determining whether linking the other element to the destination would violate a predefined rule; and

when it is determined that linking the other element to the destination would not violate a predefined rule,

modifying the corresponding code of the source to reflect the removal of the link between the source and the destination, and

adding new code to the corresponding code of the other element to reflect a new link between the other element and the destination element.

10

15

137. The computer-readable medium of claim 136, wherein the step of determining whether linking the other element to the destination would violate a predefined rule, comprises the steps of:

determining whether the other element is a class and whether the destination is another class; and

when it is determined that the other element is the class and that the destination is the other class,

identifying the new link between the other element and the destination as an inheritance link.

138. The computer-readable medium of claim 136, wherein the step of determining whether linking the other element to the destination would violate a predefined rule, comprises the steps of:

5 determining whether the other element is a class and whether the destination is an interface; and

when it is determined that the other element is the class and that the destination is the interface,

identifying the new link between the other element and the destination as an implementation link.

10 139. The computer-readable medium of claim 138, wherein the method further comprises the step of identifying a link error when it is determined that the destination is not the interface.

15 140. The computer-readable medium of claim 136, wherein the step of determining whether linking the other element to the destination would violate a predefined rule, comprises the steps of:

 determining whether the other element is an interface and the destination is another interface; and

 when it is determined that the other element is the interface and the destination is the other interface,

20 identifying the new link between the other element and the destination as an inheritance link.

141. The computer-readable medium of claim 140, wherein the method further comprises the step of identifying a link error when it is determined that the other element is not the interface.

25 142. The computer-readable medium of claim 140, wherein the method further comprises the step of identifying a link error when it is determined that the destination is not the other interface.

143. A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of elements and having a link between two of the plurality of elements, wherein each element has corresponding code and the linked elements include a source and a destination, the 5 method comprising the steps of:

displaying a graphical representation of the corresponding code of each of the plurality of elements;

receiving a selection of one of the linked elements;

receiving an identification of another of the plurality of elements that is different than the linked element;

determining whether the selected element is the destination; and

when it is determined that the selected element is the destination,

determining whether the other element is a class and whether the destination is another class; and

when it is determined that the other element is the class and that the destination is the other class,

identifying a new link from the other element to the destination as an inheritance link;

removing a portion of the corresponding code of the source that reflects the link between the source and the destination;

adding new code to corresponding code of the other element to reflect the new link between the other element and the destination;

modifying the graphical representation of the corresponding code of the source to reflect the removal of the link; and

modifying the graphical representation of the corresponding code of the other element to reflect the new link.

144. The computer-readable medium of claim 143, wherein the method further comprises the steps of:

when it is determined that the other element is the class and that the destination is not the other class,

5 determining whether the destination is an interface; and

when it is determined that the destination is the interface,

identifying the new link between the other element and the destination as an implementation link.

145. The computer-readable medium of claim 143, wherein the method further

10 comprises the steps of:

when it is determined that the other element is not the class and that the destination is not the other class,

determining whether the other element is an interface and whether the destination is another interface; and

when it is determined that the other element is the interface and that the destination is the other interface,

identifying the new link between the other element and the destination as an inheritance link.

146. The computer-readable medium of claim 143, wherein the method further comprises the steps of:

when it is determined that the selected element is the source,

determining whether the source is a class and whether the other element is another class; and

when it is determined that the source is the class and that the other element is the other class,

identifying the new link between the source and the other element as an inheritance link;

removing a portion of the corresponding code of the source that reflects the link between the source and the destination; and

adding new code to corresponding code of the source to reflect the new link between the source and the other element.

147. The computer-readable medium of claim 146, wherein the method further comprises the step of modifying a graphical representation of the corresponding code of the source to reflect the removal of the link and to reflect the new link when it is determined that linking the source to the other element would not violate a predefined rule.

148. The computer-readable medium of claim 146, wherein the method further comprises the steps of:

when it is determined that the source is the class and that the other element is not the other class,

determining whether the other element is in an interface; and

when it is determined that the other element is the interface,

identifying the new link between the source and the other element as an implementation link.

149. The computer-readable medium of claim 146, wherein the method further comprises the steps of:

when it is determined that the source is not the class and that the other element is not the other class,

5 determining whether the source is an interface and the other element is another interface; and

when it is determined that the source is the interface and the other element is the other interface,

10 identifying the new link between the source and the other element as an inheritance link.

100240-07052860

150. A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of elements, the method comprising the steps of:

receiving an identification of a first of the plurality of elements;
receiving an identification of a second of the plurality of elements;
receiving an indication that the first element is to be included in the second element;
determining whether the first element is a class and whether the second element is another class; and
when it is determined that the first element is the class and that the second element is the other class,
transferring code corresponding to the first element into the second element.

151. The computer-readable medium of claim 150, wherein the method further comprises the step of displaying a graphical representation of the code of the first element and a graphical representation of the code of the second element.

152. The computer-readable medium of claim 151, wherein the method further comprises the step of modifying the graphical representation of the code of the second element to reflect the transfer of the code corresponding to the first element into the second element.

153. The computer-readable medium of claim 150, wherein the step of transferring
20 code comprises the steps of:

removing the code corresponding to the first element from a file;
placing the code corresponding to the first element within the code corresponding to
the second element; and
deleting the file.

154. The computer-readable medium of claim 150, wherein the method further comprises the steps of:

when it is determined that the first element is the class and that the second element is not the other class.

determining whether the second element is a package; and

when it is determined that the second element is a package,

moving a file that includes code corresponding to the first element to a directory associated with the second element.

卷之三

155. A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of elements, each element having corresponding code, the method comprising the steps of:

displaying a graphical representation of the code of a first of the plurality of elements and a graphical representation of the code of a second of the plurality of elements;

receiving an indication that the first element is to be included in the second element; determining whether the first element is a class and whether the second element is another class; and

when it is determined that the first element is the class and that the second element is the other class.

transferring code corresponding to the first element into the second element;
and

modifying a graphical representation of the code of the second element to reflect the transfer of the first element into the second element.

156. The computer-readable medium of claim 155, wherein the step of transferring code comprises the steps of:

removing the code corresponding to the first element from a file;

placing the code corresponding to the first element within code corresponding to the second element; and

deleting the file.

157. A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of elements, each element having corresponding code, wherein code corresponding to a first of the plurality of elements is nested in the code corresponding to a second of the plurality of elements, the method comprising the steps of:

5 receiving an indication that the first element is to be removed from the second element;

determining whether the first element is a class and whether the second element is another class; and

10 when it is determined that the first element is the class and that the second element is the other class,

removing code corresponding to the first element from the second element.

158. The computer-readable medium of claim 157, wherein the method further comprises the step of placing the code corresponding to the first element into a file.

159. The computer-readable medium of claim 158, wherein the method further comprises the step of displaying a graphical representation of the code corresponding to the first element.

160. The computer-readable medium of claim 157, wherein the method further comprises the step of modifying a graphical representation of the code corresponding to the second element to reflect the removal of the first element from the second element.

161. The computer-readable medium of claim 157, wherein the method further comprises the steps of:

when it is determined that the first element is the class and that the second element is not the other class.

5 determining whether the second element is a package; and when it is determined that the second element is the package,

removing a first file that includes code corresponding to the first element from a directory associated with the second element to another directory.

162. A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of elements, wherein a first of the plurality of elements is nested within a second of the plurality of elements, the method comprising the steps of:

5 receiving an indication that the first element is to be removed from the second element;

 determining whether the first element is a class and whether the second element is another class; and

 when it is determined that the first element is the class and that the second element is the other class,

10 removing code corresponding to the first element from the second element;

 placing the code corresponding to the first element into a file; and

 displaying a graphical representation of the code corresponding to the first element.

15 163. The computer-readable medium of claim 162, wherein the method further comprises the step of modifying a graphical representation of the code corresponding to the second element to reflect the removal of the first element from the second element.

164. The computer-readable medium of claim 162, wherein the method further comprises the steps of:

20 when it is determined that the first element is the class and the second element is not the other class,

 determining whether the second element is a package; and

 when it is determined that the second element is the package,

25 removing a first file that includes code corresponding to the first element from a directory associated with the second element.

165. A data processing system comprising:
a secondary storage device further comprising a plurality of elements, each element
having corresponding code;
5 a memory device further comprising a program that receives a request to form a link,
that receives an indication of a first of the plurality of elements, that receives an
indication of a second of the plurality of elements, that determines whether linking
the first element to the second element would violate a predefined rule, and when it
is determined that linking the first element to the second element would not violate a
predefined rule, the program adds new code to the first element to reflect the link to
10 the second element; and
a processor for running the program.

166. The data processing system of claim 165, wherein the program further
displays a graphical representation of the code associated with the first element.

167. The data processing system of claim 166, wherein the program further
15 modifies the graphical representation of the code associated with the first element to reflect
the link to the second element.

168. The data processing system of claim 165, wherein when the program
determines whether linking the first element to the second element would violate a
predefined rule, the program determines whether the first element is a class and whether the
20 second element is another class, and when it is determined that the first element is the class
and that the second element is the other class, the program identifies the link from the first
element to the second element as an inheritance link.

169. The data processing system of claim 165, wherein when the program determines whether linking the first element to the second element would violate a predefined rule, the program determines whether the first element is a class and whether the second element is an interface, and when it is determined that the first element is the class and that the second element is the interface, the program identifies the link from the first element to the second element as an implementation link.

170. The data processing system of claim 165, wherein when the program determines whether linking the first element to second element would violate a predefined rule, the program determines whether the first element is an interface and the second element is another interface, and when it is determined that the first element is the interface and the second element is the other interface, the program identifies the link from the first element to the second element as an inheritance link.

171. A data processing system comprising:

a secondary storage device further comprising a plurality of elements and having a link between two of the plurality of elements, wherein each element has corresponding code and the linked elements include a source and a destination;

5 a memory device further comprising a program that displays a graphical representation of the corresponding code of each of the plurality of elements, that receives a selection of one of the linked elements, that receives an identification of another of the plurality of elements that is different than the linked element, that determines whether the selected element is the destination, and when it is determined that the selected element is the destination, the program determines whether the other element is a class and whether the destination is another class, and when it is determined that the other element is the class and that the destination is the other class, the program identifies a new link from the other element to the destination as an inheritance link, removes a portion of the corresponding code of the source that reflects the link between the source and the destination, adds new code to corresponding code of the other element to reflect the new link between the other element and the destination, modifies the graphical representation of the corresponding code of the source to reflect the removal of the link, and modifies the graphical representation of the corresponding code of the other element to reflect the new link; and

20 a processor for running the program.

172. The data processing system of claim 171, wherein when it is determined that the other element is the class and that the destination is not the other class, the program further determines whether the destination is an interface, and when it is determined that the destination is the interface, the program identifies the new link between the other element and the destination as an implementation link.

173. The data processing system of claim 171, wherein when it is determined that the other element is not the class and that the destination is not the other class, the program further determines whether the other element is an interface and whether the destination is another interface, and when it is determined that the other element is the interface and that the destination is the other interface, the program identifies the new link between the other element and the destination as an inheritance link.

174. The data processing system of claim 171, wherein when it is determined that the selected element is the source, the program further determines whether the source is a class and whether the other element is another class, and when it is determined that the source is the class and that the other element is the other class, the program identifies the new link between the source and the other element as an inheritance link, removes a portion of the corresponding code of the source that reflects the link between the source and the destination, and adds new code to corresponding code of the source to reflect the new link between the source and the other element.

175. The data processing system of claim 174, wherein the program further modifies a graphical representation of the corresponding code of the source to reflect the removal of the link and to reflect the new link when it is determined that linking the source to the other element would not violate a predefined rule.

176. The data processing system of claim 174, wherein when it is determined that the source is the class and that the other element is not the other class, the program further determines whether the other element is in an interface, and when it is determined that the other element is the interface, the program identifies the new link between the source and the other element as an implementation link.

177. The data processing system of claim 174, wherein when it is determined that
the source is not the class and that the other element is not the other class, the program
further determines whether the source is an interface and the other element is another
interface, and when it is determined that the source is the interface and the other element is
5 the other interface, the program identifies the new link between the source and the other
element as an inheritance link.

178. A data processing system comprising:
a secondary storage device further comprising a plurality of elements, each element
having corresponding code;
a memory device further comprising a program that displays a graphical
representation of the code of a first of the plurality of elements and a graphical
representation of the code of a second of the plurality of elements, that receives an
indication that the first element is to be included in the second element, that
determines whether the first element is a class and whether the second element is
another class, and when it is determined that the first element is the class and that the
second element is the other class, the program transfers code corresponding to the
first element into the second element, and modifies a graphical representation of the
code of the second element to reflect the transfer of the first element into the second
element; and
a processor for running the program.

179. The data processing system of claim 178, wherein when the program
transfers code, the program removes the code corresponding to the first element from a file,
places the code corresponding to the first element within code corresponding to the second
element, and deletes the file.

5

180. A data processing system comprising:
a secondary storage device further comprising a plurality of elements, wherein a first
of the plurality of elements is nested within a second of the plurality of elements;
a memory device further comprising a program that receives an indication that the
first element is to be removed from the second element, that determines whether the
first element is a class and whether the second element is another class, and when it
is determined that the first element is the class and that the second element is the
other class, the program removes code corresponding to the first element from the
second element, places the code corresponding to the first element into a file, and
displays a graphical representation of the code corresponding to the first element;
and
a processor for running the program.

10

181. The data processing system of claim 180, wherein the program further
modifies a graphical representation of the code corresponding to the second element to
reflect the removal of the first element from the second element.

15

182. The data processing system of claim 180, wherein when it is determined that
the first element is the class and the second element is not the other class, the program
further determines whether the second element is a package, and when it is determined that
the second element is the package, the program removes a first file that includes code
corresponding to the first element from a directory associated with the second element.

183. A system having a plurality of elements, each element having corresponding code, the system comprising:

means for receiving a request to form a link;

means for receiving an indication of a first of the plurality of elements;

5 means for receiving an indication of a second of the plurality of elements; and

means for adding new code to the first element to reflect the link to the second element in response to receiving the request, the indication of the first element, and the indication of the second element.